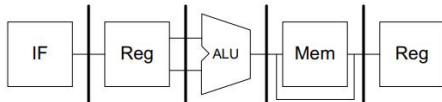


## تمرین چهارم) پاسخنامه تهیه شده توسط بابک حسینی محتشم

### تمرین دستی شماره ۴:

- ۱- یک پردازنده‌ی پایپ‌لاین با ۵ مرحله مطابق با شکل زیر در نظر بگیرید. فرض کنید تاخیر واحدها به ترتیب از چپ به راست برابر ۲، ۱، ۲، ۲ و ۱ باشد. اگر برنامه‌ای با ۱۰۰ دستور به صورت  $\text{load, add, load, add, ...}$  یکبار بر روی پیاده‌سازی تک مرحله‌ای و یکبار بر روی پیاده‌سازی پایپ‌لاین این پردازنده اجرا شود میزان تسریع پایپ‌لاین به غیر پایپ‌لاین به صورت تقریبی چقدر خواهد بود. فرض کنید که Forwarding استفاده می‌شود، هر دستور به دستور قبلی خود وابستگی داده‌ای دارد و در رجیستر فایل در نیمه‌ی اول CLK عملیات نوشتن و در نیمه‌ی دوم CLK عملیات خواندن انجام می‌شود.



با توجه به تاخیرهای داده شده، در حالت Cycle Single به کالک با دوره تناوب ns8 نیاز داریم.

$$2 + 1 + 2 + 2 + 1 = 8\text{ns}$$

در این حالت اجرای 100 دستور به  $100 \times 8 = 800\text{ns}$  نیاز دارد.

در حالت Pipeline طول کلاک به اندازه بیشترین تاخیر خواهد بود در نتیجه برای این نوع پیاده‌سازی نیاز به کلاک با دوره تناوب 2ns داریم.

چون بعد از هر دستور load دستور add می‌آید که وابستگی داده‌ای به دستور قبلی دارد پس باید پس از هر load یک سیکل پایپ را stall کنیم. البته هر load نیز با دستور قبلی خود وابستگی داده‌ای دارد که به دلیل وجود forwarding نیازی به stall کردن نیست.

در این حالت، اجرای دو دستور اول 7 کلاک زمان می‌برد و به ازای هر دو دستور دیگری 3 کلاک اضافه می‌شود. در نتیجه زمان کل انجام دستورات به شکل زیر محاسبه می‌گردد:

$$(7 + 49 \times 3) \times 2 = 308$$

میزان تسریع پیاده‌سازی پایپ‌لاین به پیاده‌سازی تک‌مرحله‌ای می‌شود:

$$\frac{800}{308} \approx 2.5974$$

۲- اگر یک Pipeline سه مرحله‌ای را به چهار مرحله‌ای تبدیل کنیم پریود از  $T$  به  $0.9T$  کاهش می‌یابد. فرض کنید 30% دستورات پرش هستند و دستور بعد از پرش وارد Pipeline نمی‌شود تا این که دستور پرش به اتمام برسد. نسبت زمان اجرای  $n$  دستور در ساختار سه مرحله‌ای به ساختار چهار مرحله‌ای چقدر است.

فرض میکنیم از  $n$  دستور،  $m$  تای آنها دستورات پرش‌اند و پایپ‌لاین ما  $k$  مرحله دارد.

برای اولین دستور  $K$  کلاک نیاز داریم. بعد از آن، اگر دستور فعلی پرش نباشد، دستور بعدی در کلاک بعدی وارد پایپ‌لاین می‌شود. اما در حالتی که دستور فعلی پرش باشد، دستور بعدی پس از  $K$  کلاک وارد پایپ‌لاین می‌شود. در نتیجه زمان اجرای  $n$  دستور با فرض اینکه نسبت دستورات پرش به کل دستورات  $m$  باشد، به شکل زیر محاسبه می‌شود:

$$T_{\text{pipeline}} = KT + mKT + (n - m - 1)T$$

( $KT$  برای دستور اول،  $mKT$  برای دستور پرش، و  $T$  برای باقی دستورات)

حالت سه مرحله‌ای  $K=3$ :

$$3T + 0.3n \times 3T + (n - 0.3n - 1)T = 1.6nT + 2T$$

حالت سه مرحله‌ای  $K=4$ :

$$4(0.9T) + 0.3n \times 4(0.9T) + (n - 0.3n - 1)(0.9T) = 1.71nT + 2.7T$$

در نتیجه نسبت زمان اجرای دستورات در حالت سه مرحله‌ای به چهار مرحله‌ای برابر است با:

$$D = \frac{1.6nT + 2T}{1.71nT + 2.7T}$$

در صورتی که تعداد دستورات زیاد باشد، حد کسر بالا به صورت زیر قابل محاسبه است:

$$\lim_{n \rightarrow \infty} D = \frac{1.6}{1.71} \approx 0.9357$$

تمرین پنجم) پاسخنامه تهیه شده توسط محمد/مانلو

۱- یک سیستم حافظه با سه سطح L1 Cache، L2 Cache و RAM را در نظر بگیرید. فرض کنید تأخیر هریک از این حافظه‌ها به ترتیب 1، 3 و 70 نانو ثانیه باشد. اگر در 90 درصد موارد دسترسی به L2 و در 100 درصد موارد دسترسی به RAM موفقیت‌آمیز باشد، نرخ برخورد (Hit Rate) حافظه L1 باید حداقل چقدر باشد که زمان دسترسی مؤثر به این سیستم حافظه از 1.5 برابر زمان دسترسی به L1 بیشتر نباشد؟

$$T_{avg} = h_1 \cdot t_1 + (1 - h_1) \cdot [h_2 \cdot (t_1 + t_2) + (1 - h_2) \cdot (t_1 + t_2 + t_3)]$$

در نهایت با جایگذاری مقادیر زیر معادله فوق را تشکیل می‌دهیم:

$$t_1=1\text{ns} \text{ و } t_2=3\text{ ns} \text{ و } t_3=70\text{ns} \text{ و } h_2=0.9 \text{ و } h_3=1$$

همچنین داریم:

$$T_{avg} = 1.5 \cdot t_1$$

در نهایت به معادله زیر خواهیم رسید:

$$h_1 \cdot 1 + (1 - h_1) \cdot [0.9 \cdot (1 + 3) + (1 - 0.9) \cdot (1 + 3 + 70)] = 1.5 \cdot 1$$

$$h_1 \cdot 1 + (1 - h_1) \cdot [3.6 + 7.4] = 1.5$$

در نهایت با حل این معادله خواهیم داشت:

$$h_1 = \frac{-9.5}{-10} = 0.95$$

۲- فرض کنید در یک برنامه ۴۰٪ از دستورات به حافظه اصلی دسترسی پیدا می کنند که از آن ۷۵٪ دستورات دسترسی به حافظه از نوع lw و ۲۵٪ باقیمانده از نوع sw باشند. فرض کنید حافظه دارای دو Cache یکسان برای دستورات و داده ها است و همچنین فرض کنید نرخ برخورد و زمان دسترسی به حافظه Cache و حافظه اصلی به ترتیب برابر ۹۸٪ ، ۱ سیکل و ۱۰۰٪ و ۲۰ سیکل باشند. در این صورت زمان دسترسی موثر به حافظه در این برنامه برای دو حالت زیر چقدر است:

**الف** - از سیاست Write Through برای نوشتن در Cache استفاده شود.

**ب** - از سیاست Write Back برای نوشتن در Cache استفاده شود. فرض کنید تا پایان برنامه هیچ خانه ای از Cache جایگزین نمی شود.

الف) ابتدای زمان دسترسی خواندن را محاسبه می کنیم:

$$T_{read} = h \cdot 1 + (1 - h) \cdot (1 + 20)$$

$$T_{read} = 0.98 \cdot 1 + 0.02 \cdot 21$$

$$T_{read} = 0.98 + 0.42 = 1.4 \text{ ns}$$

حال به محاسبه زمان دسترسی نوشتن می پردازیم: (در Write-Through ، نوشتن علاوه بر Cache به حافظه اصلی نیز ارسال می شود)

$$T_{write} = 0.98 \cdot 21 + 0.02 \cdot 21$$

$$T_{write} = 21 \text{ ns}$$

در نهایت زمان موثر دسترسی به صورت زیر محاسبه می شود:

$$1.4 \text{ ns} + 0.4 \times (0.75 \times 1.4 \text{ ns} + 0.25 \times 1.4 \text{ ns}) = 1.96 \text{ ns}$$

**ب**) حال با استفاده از سیاست Write Back محاسبات را تکرار می کنیم. به طور مشابه زمان خواندن محاسبه می شود پس داریم:

$$T_{read} = 1.4 \text{ ns}$$

برای محاسبه زمان نوشتن می دانیم در Write-Back ، نوشتن تنها در Cache انجام می شود و زمان Miss هنگام جایگزینی بلوک به حافظه اصلی ارسال می شود.

$$T_{write} = h \cdot 1 + (1 - h) \cdot (1 + 20)$$

$$T_{write} = 0.98 \cdot 1 + 0.02 \cdot 21$$

$$T_{write} = 0.98 + 0.42 = 1.4 \text{ ns}$$

در نهایت زمان دسترسی موثر را با استفاده از زمان خواندن و نوشتن محاسبه می‌کنیم.

$$1.4ns + 0.4 \times (0.75 \times 1.4ns + 0.25 \times 21ns) = 3.92ns$$